Mixed-mode Implementation of Particle Filters

Rajbabu Velmurugan*, Shyam Subramanian*, Volkan Cevher†, James H. McClellan*, and David V. Anderson*
*Georgia Institute of Technology, Atlanta, GA 30332–0250
Email: {rajbabu, shyam, jim.mcclellan, david.anderson}@ece.gatech.edu
†University of Maryland, College Park, MD 20742
Email: volkan@umiacs.umd.edu

***Abstract*— In this paper, we develop new mixed-mode implementations for particle filters and compare them to a digital implementation. The motivation for the mixed-mode implementation is to achieve low-power implementation of particle filters. The specific application considered is a bearings-only, single-target tracking algorithm. Specifically, we develop mixed-mode implementations that use analog components to realize nonlinear functions in the particle filter algorithm. The analog implementation of nonlinear functions uses low-power multiple-input translinear element (MITE) networks. Simulation results for one mixed-mode implementation of the bearings-only tracker are presented showing that analog errors are low enough to support accurate tracking. Redesign of the mixed-mode implementation in a second form with more analog components will result in nearly twenty times less power dissipation.**

I. Introduction

Particle filters or sequential Monte Carlo methods are recursive Bayesian estimation filters. Because they can handle nonlinear models and non-Gaussian noise they can be used to design more accurate estimation algorithms when compared to Kalman filter based approaches. They use discrete samples called particles and their corresponding weights to represent a distribution and provide an efficient mechanism to recursively update the distribution over time. The updated distributions can be used to obtain states. Target tracking is one such application that uses particle filters to estimate a moving target's state.

The use of large number of particles and nonlinear functions in particle filter algorithms increase their computational complexity and execution time. While most stages in the particle filter algorithm can be parallelized, the resampling stage cannot be easily parallelized. Parallelizing the various stages in the particle filter algorithm leads to faster execution time and efficient hardware architectures. This has been addressed in earlier works [1] by developing distributed resampling schemes. The work in [2] developed a fully parallel particle filter implementation that uses N processors to process N particles in the particle filter algorithm.

For certain applications, the particle filter algorithms will have to be deployed in environments where power dissipation is a constraint. Hence, we address power dissipation in particle filters. We use analog multiple-input translinear element (MITE) circuits for implementing nonlinear functions in a particle filter algorithm to reduce power dissipation. We use these analog circuits in mixed-mode implementations that use both analog and digital circuits to implement the various stages in the algorithm. A bearings-only target tracking algorithm is used as an example application to analyze the mixed-mode implementations. The results and analysis provided here are based on simulations. The contributions in this paper are an extension to our earlier work in [3].

This paper is organized as follows. Section II briefly introduces the bearings only tracking problem that uses particle filters to estimate target states. The proposed mixed-mode implementations are presented in Section III. Simulation results for the analog and mixed-mode implementations are presented in Section IV. Section V presents an analysis of the power dissipation in the mixed-mode implementation and compares it to a digital implementation and Section VI summarizes the applicability of mixed-mode implementation to particle filters.

II. Particle Filter-based Tracking

Bearings-only tracking involves estimating the target trajectory using angle measurements at a stationary sensor node. The target is assumed to move in the x-y plane and to follow a constant velocity motion model, with a state update period of 1 s. The state transition is described using the relation

$$\mathbf{X}_t = \mathbf{F}\mathbf{X}_{t-1} + \mathbf{\Gamma}\mathbf{u}_t, \tag{1}$$

where $\mathbf{X}_t = [x \quad v_x \quad y \quad v_y]_t^T$, $\mathbf{u}_t = [u_x \quad u_y]_t^T$,

$$\mathbf{F} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{\Gamma} = \begin{pmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{pmatrix}. \tag{2}$$

Here, x and y are the Cartesian coordinates of the target, v_x and v_y are the corresponding velocities. The parameter $\mathbf{u}_t$ represents the system noise and is Gaussian distributed with covariance $\mathbf{\Sigma}_u = \sigma_u^2\mathbf{I}_2$, where $\mathbf{I}_2$ is a 2×2 identity matrix. The angle measurements at a stationary sensor at the origin are given by

$$z_t = \arctan\{y_t/x_t\} + r_t, \tag{3}$$

where r_t represents a Gaussian measurement noise $\mathcal{N}(0, \sigma_r^2)$.

We use particle filters [4], [5] for solving the bearings-only tracking problem. Using (1) and (3), a particle filter algorithm for target tracking similar to the one in [5] can be formulated. The state update is used to propose new particles. This proposal function provides a sub-optimal recursive estimate of the target position in the x-y plane. The pseudo-code for the particle filter algorithm is shown in Table I.

III. Particle Filter Implementation

The complexity of particle filter algorithms depends on the number of particles N, the proposal function used, and the nonlinear functions in the model. Most applications that use particle filters perform a Gaussian evaluation at the weighting stage. Some applications might use additional nonlinear functions in the particle proposal and weight evaluation stage. In this paper, we are interested in reducing the power dissipated while performing computations. We concentrate on the nonlinear operations such as *Gaussian* and *arctan* evaluation for a single particle in the weight evaluation stage of the bearings-only tracker.

TABLE I: Bearings-only tracker particle filter pseudo-code.

```
Given the observed data $z_t$ at $t$,
  1) For $i = 1, 2, \dots, N$ sample or propose particles using the state
     update (1), $\mathbf{X}_t^{(i)} \sim p(\mathbf{X}_t^{(i)} | \mathbf{X}_{t-1}^{(i)})$.
  2) For $i = 1, 2, \dots, N$ calculate the weights,
         $\hat{w}_t^{(i)} = w_{t-1}^{(i)} p(z_t | \mathbf{X}_t^{(i)}),$    (4)
     where $p(z_t | \mathbf{X}_t^{(i)})$ is the observation density, given by
         $\frac{1}{\sqrt{2\pi{\sigma_r}^2}} \exp\left\{ - \frac{\left( z_t - \arctan \frac{y_t^{(i)}}{x_t^{(i)}} \right)^2}{2{\sigma_r}^2} \right\}.$    (5)
  3) Normalize the weights $w_t^{(i)}$ using $\hat{w}_t^{(i)}$.
  4) Calculate the state estimates, $E\{\mathbf{X}_t\} = \sum_{i=1}^{N} w_t^{(i)} \mathbf{x}_t^{(i)}$.
  5) Resample $\left\{ \mathbf{X}_t^{(i)}, w_t^{(i)} \right\}$ to obtain new set of particles
     $\left\{ \mathbf{X}_t^{(j)}, w_t^{(j)} = \frac{1}{N} \right\}$.
```

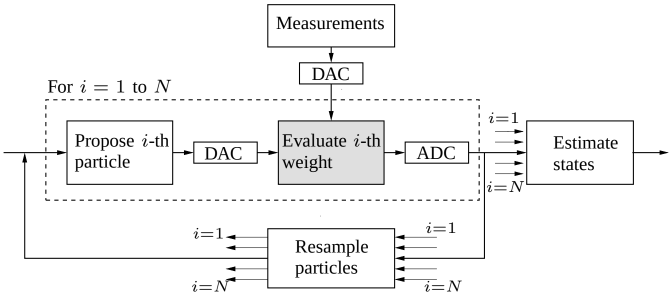


Fig. 1: Block diagram showing computational flow in the particle filter algorithm in the mixed-mode implementation (Method-1). Highlighted stage is performed in the analog domain.

A. *Mixed-mode implementations*

In the mixed-mode implementation of the particle filter algorithm, certain stages use analog components to perform the computations while the remaining stages use digital components. This leads to two possible methods that differ in their analog-digital partition.

In Method-1, shown in Fig. 1, the weight evaluation stage is implemented in the analog domain and the remaining stages in the digital domain. Data converter blocks, digital-to-analog (DAC) and analog-to-digital converters (ADC), are used to transfer data across the domains. The minimum number of bits to be used in the DAC and ADC is dictated by state-space requirements. The lowest value to be represented must be above the noise level in the analog circuit. Increasing the number of bits in these blocks will increase latency and power consumption. Hence a compromise among accuracy, speed, and power consumption has to be made such that the power savings from the analog computations is not offset.

In Method-2, shown in Fig. 2, the resampling stage alone is implemented in the digital domain and the remaining stages in the analog domain. The measurements, which are angles in the bearings-only tracker, may be from a source localization algorithm implemented in the analog domain [6]. The proposal stage uses six additions to implement (1). In the analog domain, the addition of two signals is based on Kirchoff's current law (KCL) and is performed by connecting the wires carrying the corresponding currents. In the digital domain, addition is performed using adder circuits. The state estimation, Step 4 in Table I, can be achieved by using a vector-matrix multiplier in the analog domain [7]. The resampling stage assumes the availability of an analog memory whose access is controlled using a digital controller [8].

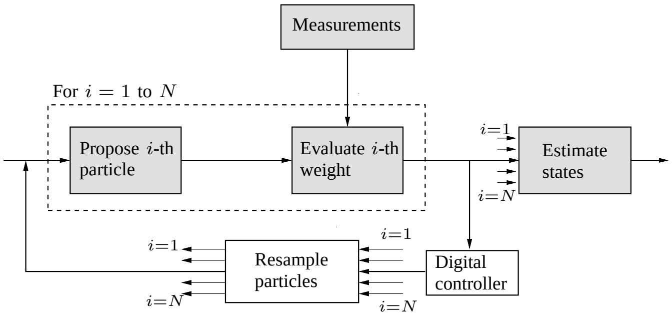


Fig. 2: Block diagram showing computational flow in the mixed-mode implementation (Method-2). Highlighted stages are performed in the analog domain.

B. *Nonlinear function realization using MITEs*

The multiple-input translinear element (MITE) was introduced in [9] as a generalization of the bipolar transistor. A 2-input MITE is defined as a circuit element satisfying the following properties:

1) The current through the input gates is zero.
2) The drain current I and the input-gate voltages V_1 and V_2 are related as $I = I_s \exp(\kappa(V_1 + V_2)/U_\mathrm{T})$, where I_s is a pre-exponential scaling constant, κ is a positive dimensionless weight, and U_T is the thermal voltage, kT/q.

MITEs operate in the subthreshold region and hence dissipate low-power. The two-input MITE is used to realize the the inverse tangent *arctan* and the *Gaussian* ($\exp(-x^2/2)$) functions in the weighting stage (5). The approximations and the corresponding implementations of these functions are considered in [3].

IV. Simulations

A. *Weight evaluation using MITEs*

Initially, the *arctan* and *Gaussian* circuits are simulated for various values of the input currents and the reference currents to determine the accuracy of the implementation. In our simulations, analog implementation of functions refers to the analog circuit model of the synthesized MITE networks [3]. Models for the AMI 0.5 μm CMOS process were used in the simulations. The results for the individual functions are shown in Table II. The error (as a percentage of the maximum) and power values correspond to the ranges of the input and reference currents shown in Table II.

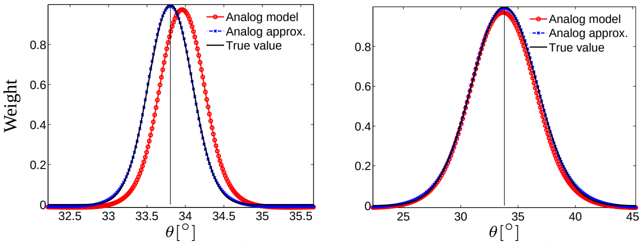
TABLE II: MITE implementation - *arctan* and *Gaussian* functions.

Circuit	*arctan*		*Gaussian*	
	Minimum	Maximum	Minimum	Maximum
Ref. I_a (nA)	0.5	20	0.1	10
Input current	$-10I_a$	$10I_a$	$-10I_a$	$10I_a$
Power (μW)	0.361	14.45	1.097	109.7
Error (%)	0.31	0.71	2.04	2.8

Next, the analog circuit model for the weighting stage of the bearings-only tracker is simulated and the results compared to a Matlab simulation. Figure 3 compares the weights obtained from the *Analog model* to the *True value* obtained using (4) and the *Analog approximation* from [3], for two cases. The error introduced by the

TABLE III: Simulation parameters - Bearings-only tracker.

Tracker parameters			**Circuit parameters** (variables in (5))		
N	σ_r	σ_u	$I_{\text{in}}(y_k^{(i)}/x_k^{(i)})$	Ia_{arctan}	$Ia_{\text{Gaussian}}(\sigma_r)$
1000	0.29 °	0.001	6.36nA to 7.18nA	10nA	0.1nA



(a) $\sigma_r = 0.29°$ (b) $\sigma_r = 2.8°$

Fig. 3: Comparison of the evaluated weights (5) using the analog (MITE) implementation for two values of measurement noise variance. The vertical line at $33.8°$ represents the true mean. (a) The standard deviation $\sigma_r = 0.29°$ is of the same order as the MITE implementation error in the *arctan* function, leading to a shift in the *Gaussian* mean. (b) The standard deviation $\sigma_r = 2.8°$ is much larger than the MITE implementation error in the *arctan* function.

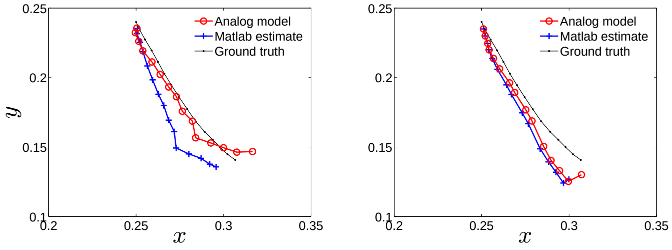
arctan block shifts the mean of the *Gaussian* distribution in the weight evaluation. Depending on the measurement noise variance σ_r^2, the shift in the Gaussian may affect the state estimate. If the variance σ_r^2 is comparable to the error in the MITE implementation of the *arctan* circuit, the error in the computation will lead to biased estimates. The influence of σ_r on the output of the weighting stage is shown in Figure 3 for two values that differ by a factor of ten.

B. Bearings-only Tracking

In this section, the weighting stage, implemented as a MITE circuit is used in a Matlab simulation of the bearings-only tracker. The tracking scenario considered here and the parameters used are similar to those in [5] and are shown in Table III. A single target trajectory and measurements that follow (1) and (3) are simulated for two different values of measurement noise. The sensor is located at the origin in the plane. The target's actual initial state is $\mathbf{X} = [0.250 \quad 0.001 \quad 0.240 \quad -0.005]^T$, the system noise is $\mathcal{N}(0, 0.001^2)$, and the measurement noise is either $\mathcal{N}(0, 0.29^2)$ or $\mathcal{N}(0, 2.8^2)$.

The Matlab simulation of the tracker is performed using scripts that execute a call to an analog (Spectre) circuit simulator tool. During each iteration of the simulation, inputs to the analog circuit were presented as a data file of $\frac{y_t}{x_t}$ quotient values and bearing measurements z_t. Here, x_t and y_t represent the x and y components of the particle state vector $\mathbf{X}$, respectively. A Spectre simulation script uses these values to compute the particle weights $\hat{w}_t^{(i)}$ using the analog MITE circuit implementation of the weighting stage. These weights are used to continue the Matlab simulation.

Tracking estimates from the *Analog model* are shown in Fig. 4. The error introduced when σ_r is small can lead to divergence of the estimates with time. Although, particle filters can mitigate this error because they use particles spread over a certain region, the error in computation might dominate and cause the particle distribution to be biased. However, if the noise variance is high compared to the error in the *arctan* computation, the estimates will be close to the ones obtained from a Matlab simulation.



(a) $\sigma_r = 0.29°$ (b) $\sigma_r = 2.8°$

Fig. 4: Comparison of x-y track estimates from the analog (MITE) implementation to the Matlab simulation. (a) for a measurement noise standard deviation of $\sigma_r = 0.29°$. (b) for a measurement noise standard deviation of $\sigma_r = 2.8°$.

The error in the *arctan* computation can vary from $0.18°$ to $0.41°$. In Fig. 4 (a) the results correspond to a standard deviation of $0.29°$. This value of the standard deviation is comparable to the MITE implementation error in the *arctan* computation. This results in a small bias in the x-y estimates, because of the shift in the mean of the *Gaussian* output, or weights, which was shown in Fig. 3 (a). Hence, the resulting state estimates (*Analog model*) do not closely match the *Matlab estimate*. However, in spite of the bias they closely follow the *Ground truth* values. This is an effect of the bias pulling the estimate towards the ground truth. The results in Fig. 4 (b) correspond to a standard deviation of $2.80°$. This noise value is high compared to the error values in the *arctan* computation and the estimated weights are not biased as shown in Fig. 3 (b). Hence, the corresponding estimates (*Analog model*) closely follow the *Matlab estimate*. The MITE implementation error in the magnitude of the *Gaussian* function output does not affect the estimates. This is because the computed weights $\hat{w}_t^{(i)}$ are normalized to obtain $w_t^{(i)}$, which are used in state estimation.

V. Power Dissipation

This section compares the power dissipated in the mixed-mode implementations to a digital ASIC implementation. From (4) and (5), note that the significant computations in the particle weighting stage of the particle filter algorithm involve the evaluation of *arctan* and *Gaussian* functions. Hence, the power consumed in these computations is considered here.

1) Analog MITE model: The analog models of the *arctan* and *Gaussian* functions use MITEs. The instantaneous power consumed by these circuits depends on the instantaneous value of the input current, I_x. As shown in Table II, for $I_a = 10$ nA with a 3 V supply, the maximum power consumed in the *arctan* circuit is 7.23 μW and the power consumed in the *Gaussian* circuit is 109.7 μW.

2) Digital implementation: The power consumption in digital CMOS circuits, ignoring the negligible static power, can be approximated [10] as, $P = \alpha_{0\to1} C_L V_{dd}^2 f_{\text{clk}}$. Here $\alpha_{0\to1}$ is the node transition activity factor, C_L the load capacitance, V_{dd} the supply voltage, and f_{clk} the clock frequency.

Digital implementations of the *arctan* and *Gaussian* functions use the CORDIC algorithm [11], which is an iterative algorithm that can be used to implement trigonometric and hyperbolic functions. The *Gaussian* function can be implemented using the relation $e^x = \sinh(x) + \cosh(x)$. A VHDL implementation of a pipelined CORDIC algorithm [12] was simulated, and the estimated power consumption using the AMI 0.35 μm CMOS process is shown in Table IV. Low-power implementation of CORDIC algorithms is an on going research area. Hence, for reference, results from a low-power CORDIC implementation using MOS Current Mode logic (MCML) demonstrated in [13] is also shown. The implementation specifications in [13], however, were a bit different.

TABLE IV: Digital implementation - CORDIC algorithm.

Implementation	Digital	MCML [13]
$\mathbf{V_{DD}}$(V)	3.3	1.0
Clock Frequency (MHz)	1	125
Output precision (bits)	12	8
Power (μW) [†a]	550×2	4330×2

[†a] Factor of 2 is for the *arctan* and *Gaussian* functions.

3) Comparison: Since the mixed-mode Method-1 uses DACs and ADCs, the power dissipation in these devices has to be considered as part of the total power dissipation in this mode. It is difficult to quantify the exact speed of operation of the MITE circuits when used in implementing Method-1. Based on the delay in the circuit their sampling frequency is in the 10 to 100 kHz range. The accuracy that can be obtained using these circuits is nearly 8 bits. Hence, we assume low speed and small wordlength for ADCs and DACs in our analysis. A TI 10-bit DAC [14] with an update rate of 75 kHz dissipates 0.75 mW, while an 8-bit ADC [15] with 70 kHz throughput has a minimum power dissipation of 0.18 mW. Using this approximate analysis, the DAC and ADC conversion dissipates nearly 1 mW power.

The MITE implementation of the *arctan* and *Gaussian* functions together dissipates nearly 120 μW (Table II). The combined digital implementation, operating at a lower frequency (Table IV) dissipates 1100 μW. This shows that the MITE implementation dissipates ten times less power than a digital implementation of the nonlinear functions considered. However, if the ADCs and DACs are included, the total power dissipated in the mixed-mode is nearly 1120 μW which is of the same order as the digital implementation.

On the other hand, if Method-2 in Figure 2 is used the power dissipated in the data converters can be removed. Implementing the proposal and state estimation stages in the analog domain can provide additional savings. While it is difficult to give exact power savings without an implementation, we can provide an estimate. The particle proposal stage involves six additions. For a digital implementation of a 16-bit ripple-carry adder operating at 2 MHz, the power dissipation is 0.21 mW (See [16], Paper 4.7). Assuming a frequency of operation in the kHz range, the power consumed will be in the μW range. In an analog implementation, addition can be achieved by connecting the wires that carry the corresponding values, and hence no additional power is dissipated. The state estimation, which is a multiply-accumulate (MAC) operation, can be performed using a vector-matrix multiplier. At frequencies of operation less than 10 MHz, when compared to a digital implementation of a MAC an analog implementation can provide a factor of 1000 in power savings [7]. For frequencies of operation in the kHz range, the analog circuit can lead to approximately a factor of ten in power savings. Hence, using the implementation in Method-2, in addition to the *arctan* and *Gaussian* functions that provide a factor of ten power savings, the proposal and state estimation stage provide a factor of nearly ten in power savings. Overall, the implementation in Method-2 will dissipate approximately 20 times less power when compared to a digital implementation. Table V summarizes the results of comparing the power dissipation in the mixed-mode methods to a digital implementation.

TABLE V: Power dissipation in bearings-only tracker.

Operation	**Mixed-mode**		**Digital**
	Method-1	Method-2	(ASIC)
Nonlinear functions	120 μW	120 μW	1100 μW
ADC/DAC	1000 μW	–	–
Addition	20 μW	–	20 μW
Multiply accumulate	$10 \times P_{\text{MAC}}$	P_{MAC} [†a]	$10 \times P_{\text{MAC}}$
Overall	1140 μW+ $10 \times P_{\text{MAC}}$	120 μW+ P_{MAC}	1120 μW+ $10 \times P_{\text{MAC}}$

[†a] P_{MAC} - power dissipation in analog multiply accumulate.

VI. Conclusions

In this paper, we presented mixed-mode implementations for particle filters to reduce power dissipation. We used analog multiple-input translinear element (MITE) networks to implement the nonlinear functions in the weight evaluation stage of a bearings-only tracker. We showed that if the MITE implementation error in the *arctan* function is smaller than the measurement noise standard deviation, its impact on the target tracking estimates will be minimal. The MITE implementation of the functions will dissipate approximately ten times less power when compared to a digital implementation of the nonlinear functions using the CORDIC algorithm. However, when the MITE implementation is used in the mixed-mode Method-1, the use of data converters can offset the power savings obtained from the analog implementation. An improved mixed-mode method that uses more analog components will lead to more power savings.

References

[1] M. Bolic, "Architectures for efficient implementation of particle filters," Ph.D. dissertation, Stony Brook University, New York, Aug 2004.

[2] S. Maskell, B. Alun-Jones, and M. Macleod, "A single instruction multiple data particle filter," in *Proc. IEEE Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, September 2006.

[3] R. Velmurugan, S. Subramanian, V. Cevher, *et al.*, "On low-power analog implementation of particle filters for target tracking," in *Proc. 14th European Signal Processing Conf. EUSIPCO*, Sep 2006.

[4] A. Doucet, N. Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[5] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approaches to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc.-F*, vol. 140, no. 2, pp. 107–113, Apr 1993.

[6] M. Stanacevic and G. Cauwenberghs, "Micropower gradient flow acoustic localizer," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 10, pp. 2148–2157, Oct 2005.

[7] R. Chawla, "Power-efficient analog systems to perform signal-processing using floating-gate MOS device for portable applications," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA, USA, Dec 2004.

[8] R. R. Harrison, J. A. Bragg, P. Hasler, *et al.*, "A CMOS programmable analog memory-cell array using floating-gate circuits," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 1, pp. 4–11, January 2001.

[9] B. A. Minch, C. Diorio, P. Hasler, *et al.*, "Translinear circuits using subthreshold floating-gate MOS transistors," *Analog Integrated Circuits and Signal Processing*, vol. 9, no. 2, pp. 167–179, 1996.

[10] A. P. Chandrakasan and R. W. Brodersen, *Low Power Digital CMOS Design*. Kluwer Academic Publishers, 1995.

[11] S. Wang, V. Piuri, and E. E. Swartzlander Jr., "A unified view of CORDIC processor design," in *Proc. IEEE 39th Midwest Symp. Circuits and Syst.*, vol. 2, 1996, pp. 852–855.

[12] Opencores, http://www.opencores.org, January 2007.

[13] J. Musicer and J. M. Rabaey, "MOS current mode logic for low power, low noise CORDIC computation in mixed-signal environments," in *Proc. Intl. Symp. Low Power Electronics and Design*, 2000, pp. 102–107.

[14] *TLC5615C, TLC5615I – 10-BIT DIGITAL-TO-ANALOG CONVERTERS*, Texas Instruments, TX, USA, Aug 2003, TI ID SLAS142D.

[15] *ADS7830 - ANALOG-TO-DIGITAL CONVERTER*, Texas Instruments, TX, USA, Mar 2005, TI ID SBAS302A.

[16] A. P. Chandrakasan and R. W. Brodersen, Eds., *Low Power CMOS Design*. IEEE Press, 1998.